APEX 新加坡 亚太交易所

APEX Trading API Development Guide

V1.1

Review History:

Version	Date	Changes
1.0	29 June 2018	First Draft
1.1	20 Dec 2018	• Added “Order Management” subsection • Added “Local Files” subsection
1.2	4 Sep 2019	Added “Order Types” section

Table of Contents

1 Objective

The objective of this guide is to provide best programming for systems that interact with APEX trading platform.

2 Development Guides

2.1 Login

- **Same login for Trading and MD**

  The same login ID can be used to login both Trading Gateway and Market Data Gateway concurrently. However, if the same login is used to login to the same gateway again, the existing login will be forced to disconnect.

- **Use Name Server for login**

  Although APEX API supports login through both Name Server and Gateway Server, only login through Name Server is advised. This is because during failover, Name Server will return the new Gateway Server IP for TraderAPI to login so that failover will be transparent for users.

  There are four Name Servers in production: two in primary and two in DR (Disaster Recovery). User should always try the two primary Name Servers first followed by the two in DR. Below is a sample of such implementation (please check with APEX on latest Name Server IPs):

  - Trading:

```
char *ns[] = {
  "tcp://10.32.100.31:4901",
  "tcp://10.32.100.32:4901",
  "tcp://10.32.100.33:4901",
  "tcp://10.32.100.34:4901"
}
for (int i = 0; i < 4; i++) {
  pTraderApi->RegisterNameServer(ns[i]);
}
```

  - Market Data:

```
char *ns[] = {
  "tcp://10.32.100.31:4903",
  "tcp://10.32.100.32:4903",
  "tcp://10.32.100.33:4903",
  "tcp://10.32.100.34:4903"
}
for (int i = 0; i < 4; i++) {
  pMdApi->RegisterNameServer(ns[i]);
}
```

- **Retry upon Login Error**

  APEX will advise on login/logout timings and users are advised to login before the suggested login time and logout after the suggested logout time. However, if the user logins too early before Trading Engine is ready, following error may be returned:

  - ErrorID = 45
  - ErrorMsg = "INVALID DATA GROUP DATASYNC STATUS IN INITIALIZATION"

  User is advised to retry login every 1 minute until Trading System is ready with successful login return code, to prevent failed login.

- **User ID Type**

  There are two types of User ID:

  - Single Trade: only able to receive order/trades initiated by current login user.
  - Manage Trade: able to receive order/trades initiated by all login users under the current Member ID and its sponsored Member ID(s).

  In production, only "Single Trade" will be issued unless "Manage Trade" is specially requested and approved.

- **Subscription Type**

  Public Topic should always be subscribed to receive instrument status updates. Single Trade User should also subscribe User Topic to receive orders and trades initiated by the user himself. On the other hand, Manage Trade User should also subscribe Private Topic to receive orders and trades initiated by all users under the same Member.

- **Local Files**

  During runtime, the API generates local files with ".con" file extension containing connection information. The path of the local files can be specified by calling *CreateFtdcTraderApi* method. Different users must specify different paths, otherwise they may not be able to receive some data from the Trading System.

2.2 Logout

Although user can call *ReqUserLogout* method to logout, API will automatically re-connect, as a mechanism to maintain connectivity. So in order to logout and disconnect, user needs to explicitly terminate the connection inside *OnRspUserLogout* method.

2.3 Disconnection handling

Upon disconnection and re-login, user needs to take care of following scenarios correctly to avoid missing or redundant data:

- If *APEX_TERT_RESTART* mode is used upon re-login, since all orders/trades since beginning of the day will be resent, user need to be able to identify and ignore duplicate orders/trades/MD (by unique identifiers like OrderSysID, OrderLocalID and TradeID).
- If *APEX_TERT_RESUME* mode is used upon relogin, and there is data received before disconnection but no processed (due to client application core dump or malfunction), there could be missing orders/trades. This mode should not be used if it is not guaranteed that all data will be processed.
- *APEX_TERT_QUICK* is not advised to use since it does not tell whole order history.

2.4 Order handling

- **Retrieve *MaxOrderLocalID* upon login**

User should retrieve *MaxOrderLocalID* after successful login in *OnRspUserLogin()*, which should be used as the base of new *OrderLocalID* and *ActionLocalID*.

- **Monotonically Increasing *OrderLocalID* and *ActionLocalID***

To increase the *OrderLocalID* every time in order messages, a recommended way is to combine member identifier, trading day, timestamp, and an increasing order number as *OrderLocalID*. *ActionLocalID* should be managed the same way.

- *OrderLocalID* and *ActionLocalID* are of String type and share the same series, which means next *OrderLocalID* or *ActionLocalID* must be greater than *Max[last OrderLocalID, last ActionLocalID]*.

- **Order Management**

Night session orders are carried over to the morning session and afternoon session of the same trading day. Within a trading day, each order can be uniquely identified by its OrderSysID. OrderSysID is reset every trading day, which means that orders from different trading days may have the same OrderSysID.

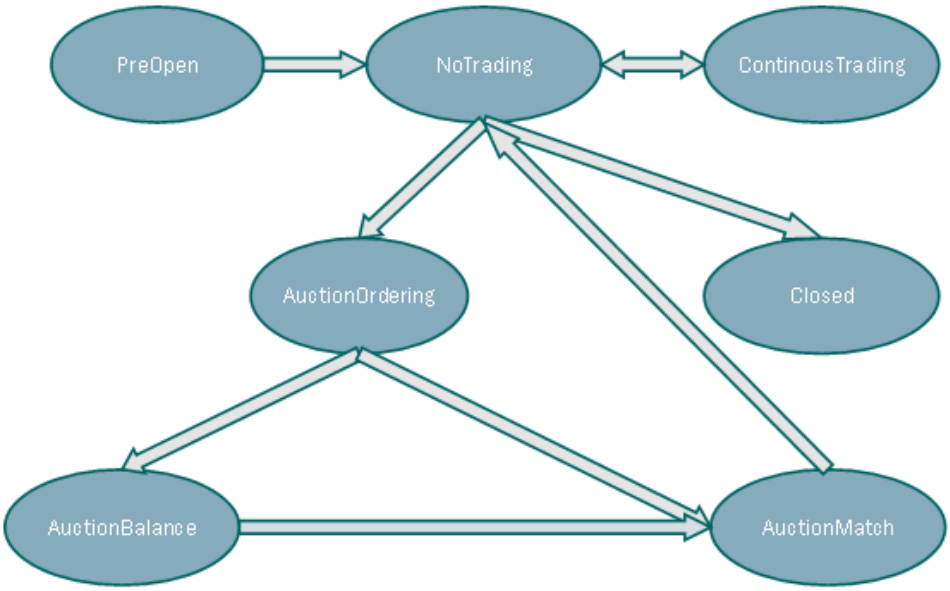
- **Order Deletion upon Market Close**

During trading pause, instrument status switches to NoTrading (`APEX_FTDC_IS_NoTrading '1'`). Orders should be kept in order book so they can resume when next session starts. At the end of trading day (normally at the end of afternoon session), instrument status switches to Closed (`APEX_FTDC_IS_Closed '6'`) and all active orders should be deleted on client. Exchange will run end of day cycle and delete all client's orders. Clients should not send API calls to delete their orders since these calls will be rejected due to Market close.

➢ **Allowed Actions for each Instrument Status**:

Instrument Status Enum	Order Allowed?	Matching Allowed?	Notes
APEX_FTDC_IS_BeforeTrading	No	No	Ready for API connection
APEX_FTDC_IS_NoTrading	No	No	Market Pause during trading sessions within a complete trading day.
APEX_FTDC_IS_Continous	Yes	Yes	Normal Trading
APEX_FTDC_IS_AuctionOrdering	Yes	No	Pre-opening
APEX_FTDC_IS_AuctionBalance	No	No	(Not in use)
APEX_FTDC_IS_AuctionMatch	No	Yes	Auction Matching
APEX_FTDC_IS_Closed	No	No	Market Close at the end of whole trading day.

➢ **Flow diagram of Instrument Status**:



2.5 Order Types

Price Condition	Volume Condition	Time Condition	Supported	Remarks
Limit Price	AllVolume	GFD	No	Unsupported order type
		IOC	Yes	FOK
	AnyVolume	GFD	Yes	Typical "GFD"
		IOC	Yes	FAK
	MinVolume	GFD	No	Volume constraint can only apply to IOC order.
		IOC	Yes	FAK with minimum filled volume
Any Price	AllVolume	GFD	No	Unsupported order type
		IOC	No	Unsupported order type
	AnyVolume	GFD	No	"Market to Limit" The remaining quantity will convert to the last price and rest in order book.
		IOC	Yes	Typical "IOC"
	MinVolume	GFD	No	Unsupported order type
		IOC	No	Unsupported order type
Best Price	AllVolume	GFD	No	Unsupported order type
		IOC	No	Unsupported order type
	AnyVolume	GFD	No	"Market to Limit" The remaining quantity will convert to the counter price and rest in order book.
		IOC	No	The remaining quantity will be cancelled.
	MinVolume	GFD	No	Unsupported order type
		IOC	No	Unsupported order type
Five Level Price	AllVolume	GFD	No	Unsupported order type
		IOC	No	Unsupported order type
	AnyVolume	GFD	No	"Market to Limit" The remaining quantity will convert to the last price and rest in order book.
		IOC	No	The remaining quantity will be cancelled.
	MinVolume	GFD	No	Unsupported order type
		IOC	No	Unsupported order type

Note: only GFD (Good For Day) and IOC (Immediate or Cancel) are available, other TimeConditions are all not effective, including:

1) GFS: Good For Session
2) GFA: Good For Auction

3) GTD: Good Till Date
4) GTC: Good Till Cancel

Only normal GFD orders are allowed in AuctionOrdering. Order entering is not allowed in AuctionMatching.

2.6 Market Data handling

- **Handling no data**

DBL_MAX is sent if a data field of floating type is empty, it should be displayed as 0 (for volume) or empty in front end. This value should be checked before further processing:

```
double format_price(const double price)
{
        return is_type_max<double>(price) ? 0.0f : price;
}
```

- **DSP**

DSP will be sent out after market close.